Data Flow Essentials Web Development

CRUD cheat sheet for building a RESTful API with Node.js And Express.Js

Handle requests to interact with db (for front end data persistence and security)

Handling User input
On Front-End use jQuery to select the element with the desired value then use an AJAX request with REST methods (GET, PUT, POST, DELETE) to send to Back-End. Use params or query string in the request URL or build a data object for a post.

jquery

```
.val()
.data()
```

Chrome network (XHR tab)
OR use Postman
POSTMAN review topics

1. 4 methods
2. Url -plus params
3. Send/ - headers
4. Response preview
5. Post body JSON

express.js

-handle front end request values with one of the following in a route (aka end point)
req.body - info from the body of the request
req.params - info in the url
req.query - other info in url after a `?q=`

EXTRA BONUS INFO USEFUL - how to dissect the incoming request URL itself
// GET 'http://www.example.com/admin/new'

```
var fullUrl = req.protocol + '://' + req.get('host') + req.originalUrl;
console.log(fullUrl);
console.log(req.protocol);
console.log(req.originalUrl);
console.log(req.baseUrl);
console.log(req.path);
console.log(req.hostname);
console.log(req.get('host'));

res.json(send json back here);
res.send(send some stuff back here);
res.render()
```

```
res.sendFile()
res.status()
res.redirect()
```

Vocab/Architecture

Controllers - where put all of our routes - we bring in models and call the ORM
Models - represents the tables in DB and helps create a relationship for controllers/db
Middleware - app.use, next()
Helper functions, configs etc DRY
Controllers lite - while models do most of logic

Cheatsheet

LANGUAGE	CREATE [POST]	READ [GET]	UPDATE [PUT]	DELETE [DELETE]
jQuery	`$.ajax({ method: "POST", url: "some.php", data: { name: "John", location: "Boston" } })` EXTRAS(beta section) - Headers - content type - Authorization	`$.ajax({ method: "GET", url: "some.php"})`	`$.ajax({ method: "PUT", url: "some.php", data: { name: "John", location: "Boston" } })`	`$.ajax({ method: "DELETE", url: "some.php", data: { name: "John", location: "Boston" } })`
Axios	`axios.post(url[, data[, config]])`	`axios.get(url[, config])`	`axios.put(url[, data[, config]])`	`axios.delete(url[, config])`
SQL	`INSERT`	`SELECT`	`UPDATE`	`DELETE`
Mongo	`db.collection.insert({})`	`db.collection.find({})`	`db.collection.update({})`	`db.collection.remove({})`
Sequelize	`db.Model.create()`	`db.Model.findAll()` `db.Model.findOne()` where	`db.Model.update()`	`db.Model.destroy()` where:{}
Mongoose	`Db.collection.create`	`Db.collection.find()`	`findOneAndUpdate`	`deleteOne()` and `deleteMany()`
Express	`app.post(...)`	`app.get(...)`	`app.put(...)`	`app.delete(...)`

Activity				
RESULT	New entry in DB	Grabs stuff from DB to send back	Edits an existing entry in the db	Deletes and entry in th db
Practical Example or use	Create new post, user, or login	Show item one at a time or all items from db on view	Edit information the user has access to	Delete a post
TIPS	Build and send a data object from the front end that usually matches or close to the db table schema			

Intermediate to Advanced Topics

Security/Authentication - api keys set up and login with passport
Nodemailer - easy free mailer for node
Cron (node-cron) - runs a function like setInterval with easier time increments setting up
TESTS!*&$# (api tests with mocha chai) - 100% coverage for sure

Key files

server.js - core file - only file to run which connects to routes and all other files
package.json - tracks npm modules and has start scripts

.env - environment variables to hid keys - KEY=value pairs - no quotes, no curly braces nothing only text syntax use process.env.WHATEVERNAME to call and use require(dotenv).config() to make ready to use - this is a file that is usually not pushed to git hub

BONUS:
Swagger.io

Debugging

Know where the code is stopping
Read VERY carefully the ENTIRE error message. Multiple times. - looking for a file path to where the code broke
If node code - hit the route with postman and hard code an object console log in the route ,

```
console.log(req.body)
console.log(req.params)
console.log("file loaded")
```

```
console.log("above function or if statement or for loop")
console.log("inside function or if statement or for loop")
```

File path to connect files is broken (esp if front end javascript or css)
Function is not being called
For loop condition or if statement condition is not really true so not running usually due to some typo or variable being undefined

```
console.log("variable name", variable) to see what the computer sees
```